AVERE
A Microsoft® Company

Cloud Cluster Management:
A vfxt.py Usage Guide

Revised 2018-07-11

Avere Systems, Inc.
www.averesystems.com

Copyright Information

The product described in this document may be protected by one or more U.S. patents, foreign patents, or pending applications.

Document Conventions

This document uses the following conventions when explaining script commands and options:

- Green monospaced text - `green` - indicates a literal term or example.
- Italic monospaced text - *`italic`* - indicates a value that should be replaced with your information.
- Red text in angle brackets - `<red>` - indicates a category of information that needs to be provided, but not the specific syntax.

Example:

```
vfxt.py --cloud-type type
        <authentication options>
        <environment options>
        --create
        --cluster-name name
        --admin-password cluster_password
        --instance-type instance_type
```

Table of Contents

Creating and Managing vFXT Clusters - About the vfxt.py Script

The vfxt.py script is a command-line tool for creating and managing Avere clusters in cloud-based virtual computing environments. The script can create new Avere clusters - including creating the vFXT nodes that make up the cluster and establishing cloud storage as core filers; destroy existing clusters (including the vFXT nodes); create and add new nodes to a cluster; and do basic cluster configuration tasks.

For ongoing cluster administration, use the Avere Control Panel. Read the Avere OS Configuration Guide for more details.

The vfxt.py script can be used with any of the cloud computing providers that Avere OS supports. Environment setup requirements are different for the different platforms, and the exact commands available vary by cloud computing provider.

This document gives a basic overview of the vfxt.py script and its options. It includes information about commands specific to Amazon Web Services and Google Compute Project cloud services. However, setting up a cloud project and configuring it to provide an Avere vFXT cluster includes many more steps than are documented here. Project creation, identity and access management, networking, quota and billing concerns, security, and many other topics are explained in detail in the Avere vFXT Installation Guide customized for your cloud provider. Read the complete details here:

- vFXT Installation Guide for Amazon Web Services
- vFXT Installation Guide for Google Cloud Platform

The command `vfxt.py --help` gives a full list of command options, including provider-specific functionality.

Installing and Setting Up the vfxt.py Script

You can run vfxt.py from any UNIX-style console environment that has access to your cloud computing resources. The simplest option is to create a Linux instance within the cloud environment, and install vfxt.py in that virtual machine. Alternatively, you can use a console from a remote Linux or Mac OS environment, as long as it has IP connectivity to your cloud instances.

Cloud Console Setup

If setting up a Linux instance in the cloud, follow these guidelines.

- Create a VM instance of any size.
- The instance must have the privileges to create other instances. Read the requirements for your cloud provider to learn how to configure the VM that provides your vfxt.py console.
- Install a recent Linux distribution from GNU, Debian, Red Hat, or CentOS. Follow the instructions in vfxt.py Software Requirements, below, for installing additional required packages.

Remote Console Setup

If using a console from a system outside the cloud environment, make sure it can access the instances within your cloud environment. Read your cloud provider's documentation to learn how to use a VPN or other utility to provide IP connectivity to your cloud instances.

Install the software described in vfxt.py Software Requirements, below.

vfxt.py Software Requirements

Before using vfxt.py, make sure that your Linux environment includes all of the necessary software packages.

1. Check security prerequisites and Python version

The vfxt.py script requires Python version 2.7.9 or later. Also, check that the system used for creating and managing the vFXT cluster meets the software requirements for securely administering an Avere cluster, as described in the appendix of the Avere OS Configuration Guide.

2. Update system dependencies

In this step, make sure that the software needed to run vfxt.py is installed and configured in the shell system. Python, SSL, and a foreign function interface (FFI) package are required. The steps are different depending on the Linux distribution:

- If using Ubuntu or another Debian-based Linux distribution:

  ```
  sudo apt-get update
  sudo apt-get install python-pip
  ```

- If using Red Hat Enterprise Linux or CentOS:

  ```
  sudo wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
  sudo rpm -ivh epel-release-latest-7.noarch.rpm
  sudo yum install -y python-pip
  ```

3. Update Python dependencies

Ensure that the Python installation has the latest versions of the API library for your cloud provider:

- Boto AWS API library if using AWS
- Google API client library if using GCE

This command can be used in either cloud provider to install the needed library.

```
pip install --user --upgrade boto requests google-api-python-client
```

Downloading and Installing vfxt.py

The vfxt.py script is published in the Python Package Index (PyPI) and also available from the Avere Systems GitHub page.

The easiest way to install vfxt.py is by using pip to automatically download and install the script and dependencies:

```
pip install --user vFXT
```

The `--user` option installs vfxt.py in `site.USER_BASE`, which defaults to `~/.local` on most UNIX-style systems. Read the Python site.py documentation to learn more.

Release archives are available from https://github.com/AvereSystems/vFXT.py/releases.

You can test that the script is active by issuing the **help** command.

```
vfxt.py --help
```

If you see the help text, the test was successful. If you do not see the vfxt.py help text, check the software requirements and try the installation again.

Configuring the Cloud Environment

Before you can use vfxt.py to create or modify Avere clusters, you must have a cloud account set up with a supported provider, and appropriate environment configuration.

Regardless of whether you use vfxt.py on a virtual machine inside your cloud environment, or from a console outside the cloud, the following basic requirements apply:

- vfxt.py must be able to make contact with the cloud provider's API endpoints (a proxy service might be required).
- vfxt.py must be able to authenticate to the cloud environment.
- vfxt.py must have permission to create and destroy virtual machine instances in the cloud environment.

The details vary greatly by cloud provider. Read the detailed Avere vFXT Installation Guide for your cloud compute provider for complete information about setting up projects, network configuration, security and privileges, and other vital information before attempting to use vfxt.py.

- vFXT Installation Guide for Amazon Web Services
- vFXT Installation Guide for Google Cloud Platform

These appendixes also give provider-specific information about using vfxt.py on particular cloud systems.

- Quick Reference - Using vfxt.py with Amazon Web Services
- Quick Reference - Using vfxt.py with Google Cloud Platform

Using vfxt.py

This section describes the syntax and options available when using vfxt.py as a command-line interface to create and modify Avere clusters.

Frequently used options are described in detail in this section, but all options are explained in the vfxt.py Command Syntax and Options section at the end of this document.

Basic Syntax

The basic form for a vfxt.py command is this:

```
vfxt.py --cloud-type type <authentication> <environment options> --<action>
```

Each vfxt.py command (except for information queries like `--help`) must include:

- The cloud provider parameter - that is, `gce` for Google or `aws` for Amazon.
- Authentication credentials for accessing the cloud API. The syntax is different depending on the cloud provider and authentication method you are using; read the setup section for GCE or AWS for details.
- The environment options for the cloud network or project (depending on the cloud provider).
- An action to perform - for example, `--create` to create a new vFXT cluster.

Note that similar commands have different names if they are restricted to one type of cloud provider. For example, the option to set an Amazon location is `region` and the option for setting a Google location is `zone`. This difference serves as a check that the correct cloud service was specified.

Actions include things like creating or destroying a cluster, adding nodes, and stopping or starting the cluster. The command `vfxt.py --help` lists all options.

Creating a Cluster

Use the `--create` action to instantiate a new vFXT cluster.

```
vfxt.py --cloud-type type
        <authentication options>
        <environment options>
        --create
        --cluster-name cluster_name
        --admin-password cluster_password
        --instance-type instance_type
```

You must supply the following parameters for a create operation:

- `--cluster-name` - The name for the new cluster.
- `--admin-password` *`cluster_password`* - The password for administering the cluster settings. This password is used to manage the cluster (either by logging into the Avere Control Panel or through the XML-RPC command-line interface) and also is required to modify or destroy the cluster with vfxt.py.
- `--instance-type` *`type`* `-` The type of cloud VM to use for each vFXT node. Check the documentation from your cloud provider to learn about instance types on your platform, and refer to the Avere vFXT Installation Guide for your cloud provider to understand the minimum instance type that can be used to create a vFXT node. An instance type that meets the minimum requirements will give

better cost management, but a more powerful instance type will give better performance - choose the supported instance type that best meets your needs.

Initial Configuration for the New Cluster

Use these options with the **create** command to set up the basic parameters for your new cluster.

- `--nodes` *`number_of_nodes`* - The number of vFXT nodes to create in the cluster. The minimum required number of nodes is 3 (this is also the default value).

- Cache size options - There are two ways to set the cluster cache size; choose one.

    - `--node-cache-size` *`size_in_GB`* - The size of the data cache for each node.
      If you use this option, the system automatically sets the number of data disks per node and the disk size. If you prefer to set these values individually, use the data-disk-count and data-disk-size options instead of using node-cache-size.

    - `--data-disk-count` *`number_of_disks`* and `--data-disk-size` *`size_in_GB`* -
      These options let you specify the number of virtual data storage disks to create for each node (count) and the size of each disk (size). If using Google Cloud, be careful not to exceed your storage quota or other limits, which would cause the create command to return an error.

      Each node in the cluster will have the same number of data disks, and each disk will be the same size.

- `--data-disk-type` *`volume_type`* - The kind of data volume to use as vFXT node disks. Values depend on the cloud provider type:
    - For GCE, options are `pd-ssd`, or `local-ssd`
    - For AWS EC2, options are `gp2` (the default), or `io1`

- `--cluster-proxy-uri` *`cluster_proxy_URL`* - Address of a proxy server to set for the cluster. (Avere does not require using a proxy server.) Use the format `http://`*`username`*`:`*`password`*`@`*`proxy_IP address`*`:`*`port`*`/` The port value is optional.

  Example: `--cluster-proxy-uri http://admin1:mygoodpw42@203.0.113.29:8080/`

  It is better to specify an IP address instead of a hostname so that the proxy server will work even if DNS is not reachable.

- `--cluster-range` *`IP_range`* - An IP address range, in CIDR notation, that the cluster will use for client-facing IP addresses and for cluster management. If the range is not specified, the cloud provider assigns a range based on the particular environment configuration and current resource utilization.

- `--vserver` *`vserver_name`* - The name to use for the cluster vserver. If not specified, the default name is "vserver". The vfxt.py **create** command gives one vserver per cluster. If you want to add vservers, use the Avere Control Panel or the XML-RPC API after creating the cluster.

- `--core-filer` *`core_filer_name`* - The name to use for the core filer being created. If not specified, the default name is the name of the cloud service type (`aws` or `gce`), or `nfs` if you specified a different NFS core filer.

- `--bucket` *`existing_bucket_name`* - Specify an empty bucket to use for the core filer.

Additional Create Settings

This section describes additional options that can be useful when creating a new vFXT cluster.

Skip Cleanup

If there is an error, vfxt.py rolls back what was done. In some situations, you might want to prevent this rollback - for example, during troubleshooting. The `--skip-cleanup` option leaves nodes, buckets, routes, roles, and other entities in the state they had when the error occurred. Anything created during the operation is not removed.

```
vfxt.py --cloud-type type
        <authentication options>
        <environment options>
        <action>
        --skip-cleanup
```

No Core Filer

vfxt.py can skip the creation of a bucket and the associated cloud core filer configuration. Use this option if you want to create a vFXT cluster and configure its storage afterward by using the Avere Control Panel or the XML-RPC command-line cluster configuration API.

```
vfxt.py --cloud-type type
        <authentication options>
        <environment options>
        <action>
        --no-corefiler
```

NFS Core Filer

vfxt.py can configure an NFS core filer at cluster creation time by providing the NFS mount point in the *`host`*`:/`*`path`* format. Note that if you specify an NFS core filer, vfxt.py does not create a cloud core filer.

```
vfxt.py --cloud-type type
        <authentication options>
        <environment options>
        <action>
        --nfs-mount mount_point_host:/path
```

Cache Disk Sizes

Data disks for the vFXT cache sizes can be configured independently with `--data-disk-size` and `--data-disk-count` at cluster creation time. A convenience option, `--node-cache-size`, automatically sizes these based on the given cache size (in GB).

```
vfxt.py --cloud-type type
        <authentication options>
        <environment options>
        --create
        --node-cache-size cache_size_per_node_in_GB
```

Cluster Management Actions

This section gives detailed descriptions of commonly used cluster management commands.

Specifying Which Cluster To Modify

Note that you must identify the cluster nodes when configuring an existing cluster with vfxt.py. There are two main ways to identify the cluster instances, depending on whether the cluster is running (online) or stopped (offline).

- For a running cluster, supply the same credentials you would use to manage a cluster from the Avere Control Panel:
    - The cluster's management address (an IP address guaranteed to be held by one of the cluster nodes for administrative access). Use `--management-address` to supply the IP address.
    - The cluster's administrative password (you set this when you create the cluster). Use `--admin-password` to supply the password.

  The vfxt.py script uses this information to query the cluster and identify instances for each cluster node.

- For actions on a stopped cluster, you must specify the cloud instance identifiers for the cluster nodes. (When a cluster is offline, you cannot query it using the management address and password.) Use the option `--instances` to provide the instance identifiers of each cluster node.

  You can find instance identifiers from your cloud administrative console. The format varies by cloud provider; some providers use unique strings and others use a URL-style format.

Add Nodes to a Cluster

The `--add-nodes` option extends the cluster.

Use the `--nodes` option to specify how many nodes to add. The cluster must be online.

```
vfxt.py --cloud-type type
        <authentication options>
        <environment options>
        --add-nodes
        --nodes number_of_nodes_to_add
        --management-address cluster_mgmt_IP_address
        --admin-password cluster_password
        [<node options>]
```

There is no restriction on the number of nodes to add, but testing has shown that adding nodes in small batches (one to three at a time) runs more quickly and is less likely to encounter trouble than adding larger batches of nodes at once.

New nodes will be identical to the existing cluster nodes unless you include options to customize them. You can use the following node options from the `--create` command with the `--add-nodes` command:

- `--data-disk-count` *number_of_disks* and `--data-disk-size` *size_in_GB*
- `--data-disk-type` *volume_type*
- `--instance-type` *type*

Read Initial Configuration for the New Cluster for details about these options.

Destroy a Cluster

The `--destroy` option permanently removes a cluster.

```
vfxt.py --cloud-type type
        <authentication options>
        <environment options>
        --destroy
        --management-address cluster_mgmt_IP_address
        --admin-password cluster_password
```

If the cluster is offline, you must provide the node instance identifiers since they cannot be discovered from the cluster configuration. You do not need to provide the management address and password for an offline cluster. See Specifying Which Cluster To Modify for more information on identifying your cluster node instances.

Stop a Cluster

The `--stop` option takes a cluster out of service. A stopped cluster does not serve client requests or update stored data. Stopping the cluster also stops its cloud virtual machines so that they do not incur usage charges; however, disk usage and storage charges can still accumulate.

```
vfxt.py --cloud-type type
        <authentication options>
        <environment options>
        --stop
        --management-address cluster_mgmt_IP_address
        --admin-password cluster_password
```

Start a Cluster

Restart a stopped cluster with the option `--start`.

The system cannot query a stopped cluster for the node list, so you must provide a list of instance identifiers for the cluster nodes.

```
vfxt.py --cloud-type type
        <authentication options>
        <environment options>
        --start
        --instances instance1_ID instance2_ID instance3_ID
```

Separate the instance identifiers with spaces.

The format of the identifiers depends on the cloud vendor, because different cloud providers have written their APIs to use different keys for instance lookup.

- AWS uses a numeric instance ID in the form `i-xxxxxxxxxxxxxxxxx`
- GCE uses an instance name, which is the text string you used to name the cluster

Proxy Configuration - API Commands

There are two different proxy configuration options in vfxt.py:

- `--proxy-uri`, to send API commands through a proxy server
- `--cluster-proxy-uri`, to set the vFXT cluster's proxy server.

This section describes the `--proxy-uri` option, which affects API commands.

To configure vfxt.py to issue cloud API calls through a proxy server, use the `--proxy-uri` setting. The proxy argument must be used on each command that you want to send over the proxy.

```
vfxt.py --cloud-type type
        <authentication options>
        <environment options>
        <action>
        --proxy-uri API_proxy_URL
```

Specify the proxy server in this format: `http://username:password@address:port_number`

You can use either an IP address or a hostname in the API proxy address value. (For the cluster proxy address, an IP address is preferred to avoid reliance on a domain name lookup service.)

Proxy Configuration - Cluster Proxy

There are two different proxy configuration options in vfxt.py:

- `--proxy-uri` to send API commands through a proxy server
- `--cluster-proxy-uri` to set the vFXT cluster's proxy server

This section describes the `--cluster-proxy-uri` option, which affects the configuration of the vFXT cluster.

```
vfxt.py --cloud-type type
        <authentication options>
        <environment options>
        <action>
        --cluster-proxy-uri cluster_proxy_URL
```

Specify the proxy server in this format: `http://username:password@IP_address:port_number`
Example: `--cluster-proxy-uri http://vfxtcluster:goodpw@203.0.113.76:8080/`

Interactive Mode

The vfxt.py script can be used in a Python interactive session by passing the `--interact` parameter. Interactive mode can be useful for API testing, or for validating authentication and environment options. The `--interact` option is a simple command-line switch that initializes a service object with the vfxt.py command line options and allows you to inspect it or run code within an interactive session.

```
vfxt.py --cloud-type type
        <authentication options>
        <environment options>
        --interact
```

An example session:

```
# ./vfxt.py --cloud-type gce --on-instance --interact

--- Service object available as 'service' ---

>>> service.export()
{'subnetwork_id': None, 'zone': ['us-central1-a'], 'network_id': 'gce1',
'access_token':
u'ya29.Cj4ZAxwtZZZQP8MJ6Z6Pf_uZZZZZZiwARn291ekyz_igBntzZzPx4BchAbmsoJ84XpNiWRrp0cjy66Hn
sFZDw', 'client_email': 'projectid-compute@developer.gserviceaccount.com',
'project_id': 'some-project'}
>>> help(service)
```

vfxt.py Command Syntax and Options

This section is adapted from the vfxt.py help file, retrieved in May 2017. Options might have changed since this document was created; use the command `vfxt.py --help` to see accurate options for your version of the script.

Required Parameters

For all operations (except the **help** and **version** commands), these elements are required:

- `--cloud-type` {`aws`|`gce`} - The cloud provider being used
- Authentication information (varies by provider)
- Environment information (varies by provider)
- Action (create, destroy, stop, start, add nodes)

For operations on existing vFXT clusters, you must supply information to identify the cluster instances.

- For an online cluster, use `--management-address` and `--admin-password` to identify the cluster being modified and authorize changes.
- For an offline cluster, use `--instances` to supply the instance identifier for each node in the cluster.

Script Information Options

These options do not require any other parameters.

`-h, --help`	Show a help message and exit
`-v, --version`	Show the program's version number and exit

Script Behavior Options

These arguments affect how vfxt.py behaves.

`-d, --debug`	Give verbose feedback
`--interact`	Use Python interactive mode
`--proxy-uri` *URL*	Send vfxt.py calls through this proxy server. Include the username, password, and port - for example, `http://user:password@172.16.16.20:8080/`
`--skip-cleanup`	Do not remove resources (buckets, volumes, instances) created as part of the action when a failure occurs
`--wait-for-state {red|yellow|green}`	After issuing the command, wait until the cluster has reached a particular state when validating success of the action. By default, vfxt.py waits until the cluster has only non-critical alerts (yellow status, based on the alert conventions used in the Avere Control Panel dashboard). The waiting system shows a status message like `INFO - Waiting for cluster healthcheck`. You should

	use this option only when troubleshooting a problem under the guidance of Avere Global Services.

Cluster Actions

These options make up the basic actions for cluster creation and maintenance.

`--create`	Create a new cluster. Read Creating a Cluster for details.
`--destroy`	Permanently remove a cluster. Read Destroy a Cluster for details.
`--stop`	Take a cluster offline. Read Stop a Cluster for details.
`--start`	Reactivate an offline cluster. Read Start a Cluster for details.
`--add-nodes`	Add nodes to a cluster. Read Add Nodes to a Cluster for details.

Cluster Configuration Options

These options apply to any supported cloud provider.

Authentication and Environment Options

`--on-instance`	Query the cloud environment for instance credentials. Use this option when running vfxt.py in a cloud instance instead of passing authentication credentials. Read the setup information for your cloud platform to learn more.
`--from-environment`	Query the local host configuration for service credentials. Use this option when running vfxt.py from a non-cloud host where you have installed your cloud provider's custom command-line tool (that is, gcloud or the AWS CLI) and configured it with the appropriate credentials. Read the CLI setup information for your cloud platform to learn more.
`--admin-password` *cluster_password*	Administrator password for cluster management. When creating a cluster, this option sets the password. When modifying an online cluster, use this option to supply the password in order to authenticate to the cluster. (If you need to change the cluster management password, use the Avere Control Panel web interface or the XML-RPC command-line API.)
`--management-address` *cluster_mgmt_IP*	The cluster's management address. When creating a cluster, this option sets the management IP address. When modifying an online cluster, use this option to specify which cluster is being modified.

`--instances` *`instance1_ID instance2_ID instance3_ID`*	Instance identifiers for the cluster nodes - use this to specify an offline cluster. Read Specifying Which Cluster To Modify for more information.

`--s3-access-key` *`S3_access_key`*
`--s3-secret-key` *`S3_secret_key`*
`--s3-profile` *`S3_profile`*
	Use these options if you need to pass an additional S3 credential for your cluster resources.

`--ssh-key` *`key`*	SSH key for cluster authentication - this option allows for systems that use key-based authentication via SSH instead of password-based authentication. For Amazon Web Services, provide the SSH key name; for Google Compute, provide the path to the public key file.

Miscellaneous Cluster Options

`--cluster-name` *`cluster_name`*	Name for the cluster (also used to tag resources). This name should be compatible with DNS, since some cloud providers process it into a DNS hostname.
`--cluster-range` *`IP_range`*	IP address range (CIDR format) for the cluster. The cluster detects unused addresses in this range and uses them for client traffic as well as cluster management tasks.
`--cluster-proxy-uri` *`URL`*	Proxy resource for the cluster - for example, `http://user:pass@172.16.16.20:8080/`. **Note:** Use the IP address rather than hostname in case DNS becomes unreachable.
`--junction` *`vserver_junction_path`*	Sets the GNS path for the vserver's junction. The path must start with "`/`". If not set, the default value is the cloud provider name (`/gce` or `/aws`), or the last segment of the NFS export path (`/smithj` for an NFS export with the path `/files/smithj`)
`--root-size` *`boot_disk_size_in_GB`*	Use this to specify the size of each node's boot disk, in GB. **Note:** This setting might not be honored by some cloud providers.
`--trace-level` *`level`*	Trace level for the created cluster
`--vserver` *`vserver_name`*	Name for the vserver that will be created with the cluster

Node and Node Cache Options

`--nodes` *number_of_nodes*	Number of nodes to create or add to the cluster.
`--node-cache-size` *size_in_GB*	Size of the data cache space for each node (in GB). Use this to automatically set `--data-disk-count` and `--data-disk-size`, or use those two settings to set each manually. All nodes will have identical storage capacity. Read Initial Configuration for the New Cluster for additional information.
`--data-disk-count` *number_of_disks*	Number of data disk volumes per cluster node. This option can be used with `--create` or with `--add-nodes`. You can use `--node-cache-size` to set this automatically. Read IInitial Configuration for the New Cluster for additional information.
`--data-disk-size` *size_in_GB*	Size of data disk volumes to create for the vFXT cluster. This option can be used with `--create` or with `--add-nodes`. You can use `--node-cache-size` to set this automatically. Read Initial Configuration for the New Cluster for additional information.
`--data-disk-type` *volume_type*	Type of storage volumes to create for the vFXT cluster cache. Values depend on the cloud provider; AWS values are `gp2` (default), `io1`, or `standard`. GCE values are `pd-ssd` (default), or `local-ssd`.
`--instance-type` *instance_type*	Type of instance to use when creating nodes. This is required for creating a cluster or when adding nodes to a cluster. Read Creating a Cluster for details.
`--image-id` *image_ID_or_URI*	Optionally, use this parameter to specify an image instead of using the default image when creating the cluster. Consult Avere Global Services for guidance before using this advanced option. The image ID or URL should match what your cloud provider uses. ● AWS image ID example: `ami-ff6e9c9f` ● GCE image URL example: `https://www.googleapis.com/compute/v1/projects/tribal-parsec-845/global/images/avere-vfxt-4625`
`--join-instance-address`	Join nodes using the instance address rather than the management address

Core Filer Options

`--core-filer` *`core_filer_name`*	Name for the core filer that will be created with the cluster.
`--bucket` *`storage_bucket_name`*	Name of an existing S3 bucket to use as the core filer (the bucket must be empty).
`--no-corefiler`	Skip creating a cloud core filer when creating the cluster. This will create a cluster without any core filers.
`--nfs-mount` *`host:/path`*	NFS mountpoint to use as the core filer (in host:/path format). If you use this option when creating a cluster, it will use the specified resource instead of creating a cloud core filer.
`--subdir` *`path_under_mountpoint`*	Use this option with `--nfs-mount` to mount a subdirectory on the NFS storage system.

AWS Options

These options can be used only if the `--cloud-type` value is `aws`.

AWS Authentication Options

`--access-key` *AWS_access_key*	AWS access key (required for off-instance authentication)
`--secret-key` *AWS_secret_key*	AWS secret key (required for off-instance authentication)

For authentication from within a cloud console on the AWS instance, use the generic option `--on-instance` instead of these keys.

AWS Environment Options

`--region` *AWS_region*	AWS region in which to create the cluster
`--subnet` *subnet_name* `[...]`	One or more subnet names, separated by spaces (for example, `subnet-xxxx subnet-yyyy`)

`--govcloud`	Set additional environment requirements for AWS GovCloud instances. This option sets the following default values:
	`--iam-host iam.us.gov.amazonaws.com`
	`--arn aws-us-gov`

Additional AWS-Specific Options

`--arn` *ARN_string*	Override the default Amazon Resource Name
`--aws-tag` *key:value* `[...]`	Key:value pairs to be added as tags. To add multiple tags, use the option multiple times (example: `--aws-tag dept:finance -aws-tag sec:124`)
`--data-disk-iops` *IOPS_count*	Number of sustained IOPS (for volume type io1)
`--ephemeral`	Use EC2 ephemeral disks for cache (**Warning:** Ephemeral disks increase the risk of data loss.)
`--iam-role` *IAM_role*	IAM role to assign to the cluster
`--iam-host` *IAM_host*	IAM host
`--no-disk-encryption`	Disable use of encryption with data disks
`--no-ebs-optimized`	Disable use of EBS optimization
`--placement-group` *group_name*	Name of a placement group to use
`--profile` *profile_name*	User profile to use when connecting to EC2/VPC/IAM

`--security-group` *`group_ID`*	Security group ID for the cluster (for example, `sg-xxxx`); separate multiple values with spaces

GCE Options

These options can be used only if the `--cloud-type` value is `gce`.

GCE Authentication Options

`--key-file` *`key_file_path`*	OATH2 service account P12 (deprecated) or JSON key file for off-instance authentication. For P12 keys, you must also provide the client email address with `--client-email`
`--client-email` *`email_address`*	OATH2 service account client email for p12 key files (older option for off-instance authentication)

Use the generic option `--on-instance` for local console commands.

GCE Environment Options

`--network` *`network_name`*	Network name
`--subnetwork` *`subnetwork_name`*	Subnetwork name
`--zone` *`GCE_zone`* `[`*`zone2`* `...]`	One or more zone names (example: `us-central1-a`). Separate multiple names with spaces.

Additional GCE-Specific Options

`--data-disk-nvme`	Use the NVME interface instead of SCSI. This option applies only to the GCE disk type `local-ssd`.
`--project` *`project_name`*	Google Cloud project name
`--local-ssd`	Use local-ssd disks for cache (**Warning:** This disk type increases the likelihood of data loss.)
`--metadata` *`key1:value1 key2:value2 ...`*	Set one or more metadata values (similar to tags) on the instance.
`--gce-tag` *`GCE_instance_tag`*	GCE instance tag

All Options

```
vfxt.py - Create an Create an Avere vFXT cluster

usage: vfxt.py [-h] [-v]
               (--create|--destroy|--stop|--start|--add-nodes|--upgrade|--interact)
               --cloud-type {aws,gce} [--s3-access-key S3_ACCESS_KEY]
               [--s3-secret-key S3_SECRET_KEY] [--s3-profile S3_PROFILE]
               [--on-instance] [--from-environment] [--image-id IMAGE_ID]
               [--access-key ACCESS_KEY] [--secret-key SECRET_KEY] [--profile PROFILE]
               [--security-group SECURITY_GROUP] [--region REGION]
               [--iam-role IAM_ROLE] [--iam-host IAM_HOST] [--arn ARN]
               [--ephemeral] [--placement-group PLACEMENT_GROUP]
               [--subnet SUBNET [SUBNET ...]] [--aws-tag AWS_TAG] [--govcloud]
               [--no-disk-encryption] [--no-ebs-optimized]
               [--client-email CLIENT_EMAIL]
               [--project PROJECT] [--zone ZONE [ZONE ...]]
               [--network NETWORK] [--subnetwork SUBNETWORK]
               [--key-file KEY_FILE] [--local-ssd] [--metadata METADATA]
               [--gce-tag GCE_TAG] [--subscription-id SUBSCRIPTION_ID]
               [--client-id CLIENT_ID] [--client-secret CLIENT_SECRET]
               [--storage-account STORAGE_ACCOUNT]
               [--premium-storage-account PREMIUM_STORAGE_ACCOUNT]
               [--instance-client-id INSTANCE_CLIENT_ID]
               [--instance-client-secret INSTANCE_CLIENT_SECRET]
               [--instance-service-principal-id INSTANCE_SERVICE_PRINCIPAL_ID]
               [--location LOCATION] [-d]
               [--skip-cleanup] [--wait-for-state {red,yellow,green}]
               [--proxy-uri URL] [--ssh-key SSH_KEY]
               [--cluster-name CLUSTER_NAME]
               [--instances INSTANCES [INSTANCES ...]]
               [--instance-type INSTANCE_TYPE]
               [--admin-password ADMIN_PASSWORD]
               [--management-address IP_ADDR] [--nodes NODES]
               [--node-cache-size NODE_CACHE_SIZE]
               [--data-disk-count DATA_DISK_COUNT]
               [--data-disk-type DATA_DISK_TYPE]
               [--data-disk-iops DATA_DISK_IOPS] [--data-disk-nvme]
               [--data-disk-size DATA_DISK_SIZE] [--root-size ROOT_SIZE]
               [--configuration-expiration CONFIGURATION_EXPIRATION]
               [--upgrade-url UPGRADE_URL] [--cluster-range CLUSTER_RANGE]
               [--cluster-proxy-uri URL] [--trace-level TRACE_LEVEL]
               [--join-instance-address] [--no-corefiler] [--bucket BUCKET]
               [--nfs-mount NFS_MOUNT] [--core-filer CORE_FILER]
               [--subdir SUBDIR] [--junction JUNCTION] [--vserver VSERVER]
```

Quick Reference - Using vfxt.py with Amazon Web Services

This section gives an overview of how to configure a vfxt.py installation to be able to create Avere clusters in an AWS Elastic Compute Cloud (EC2) environment.

Note: Read the vFXT Installation Guide for Amazon Web Services for detailed information about setting up an Amazon cloud computing environment to create vFXT clusters.

Configuring the AWS environment to give vfxt.py the necessary access includes the following steps. Refer to the AWS vFXT Installation Guide for complete information about how to set up these prerequisites:

- Ensure that your account has sufficient resources and request a quota increase if needed. (Note that quota increase requests might take a day or more to be fulfilled.)
- Send your AWS account number to Avere Systems to gain access to the vFXT image.
- Create a virtual private cloud (VPC) configured with a subnet.
- Establish an appropriate IAM policy for creating vFXT clusters.
- Associate the role with the vfxt.py user:
    - If using a cloud instance local to your cloud environment to run vfxt.py, create the VM with the cluster management IAM role.
    - If using a remote system to run vfxt.py, create a user with that role, and establish the user’s access key and secret key pair. When beginning a vfxt.py session from a remote console, pass the user keys to the cloud system to authenticate the system.
- On the console where you will use vfxt.py to run commands, install the Python Boto library to allow Python to interact with the AWS API.

Instance Requirements for running vfxt.py in AWS

If using a cloud-based VM to run the vfxt.py script, you can use any instance type that can run a Linux operating system for command-line access.

The instance must be created with the IAM role for cluster creation. Read Create the Cluster Management Role for more information.

Install the AWS SDK for Python

On the system where you will run vfxt.py, install the Python Boto library.

```
sudo pip install --upgrade boto
```

Create the Cluster Management Role

In your AWS environment, you must create a role with a policy that allows the associated users to manipulate instances, storage, and roles, among other things. vfxt.py will connect to your private cloud as a user with this role.

There are multiple steps. First, you must create a policy that enables access. Second, create a role that uses that policy. Third, create a user that is associated with that role. These settings are made in the AWS Management Console. Read the AWS vFXT installation guide for the details of each step.

Make sure the policy has settings like the following.

```
{
    "Statement": [
        {
            "Resource": "*",
            "Action": [
                "ec2:Describe*",
                "ec2:RunInstances",
                "ec2:TerminateInstances",
                "ec2:RebootInstances",
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:CreateTags",
                "ec2:DeleteTags",
                "ec2:ModifyInstanceAttribute",
                "ec2:CreateVolume",
                "ec2:DeleteVolume",
                "ec2:AttachVolume",
                "ec2:DetachVolume",
                "ec2:CreateSnapshot",
                "ec2:DeleteSnapshot",
                "ec2:RegisterImage",
                "ec2:DeregisterImage",
                "ec2:CreateImage",
                "ec2:DeleteRoute",
                "s3:CreateBucket",
                "s3:DeleteBucket",
                "s3:SetTag",
                "s3:ListBucket",
                "iam:AddRoleToInstanceProfile",
                "iam:CreateInstanceProfile",
                "iam:CreateRole",
                "iam:DeleteInstanceProfile",
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:ListRolePolicies",
                "iam:GetInstanceProfile",
                "iam:PutRolePolicy",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:PassRole"
            ],
            "Effect": "Allow"
        }
    ],
    "Version": "2012-10-17"
}
```

This example is the most general case; if your organization has defined more specific policy statements, you can use those instead, as long as the resulting permissions are similar.

Create a User with the Cluster Manager Role

If you will use vfxt.py from a remote cluster (that is, from a machine outside the AWS VPC that hosts your cluster), you must create a user that has the ability to create clusters. When using vfxt.py, you must pass that user's credentials to AWS to authorize vfxt.py to create or modify vFXT clusters.

This step is not required if using vfxt.py from an on-instance command line.

There are several ways to associate a user with a role or policy, and the best method for your situation depends on what other infrastructure you have set up in your AWS system. One method is to add the user to a group with the needed policy. Consult the AWS IAM User Guide for details.

After creating the user and assigning appropriate permissions, copy out the user credentials to use to authorize with vfxt.py commands. In the IAM console, click the **Encryption Settings** button to create and manage user credentials. Create at least one access key ID and secret key value pair to use with vfxt.py commands.

AWS Authentication Options

When executing a vfxt.py command, you must provide the user credentials defined above to authenticate to the cloud environment. The process is slightly different if running in a local cloud console or if using a remote system.

Local Cloud Instance

To run vfxt.py on a VM in the same cloud environment where you will be creating clusters, use the AWS EC2 service to create a new instance. When creating the instance, you can use any type. Make sure to specify the IAM role created above before starting the instance.

Detailed instructions for creating an instance with the cluster manager role are included in the vFXT Installation Guide for Amazon Web Services.

After creating the instance, you can pass authentication credentials from vfxt.py by querying the instance metadata to obtain an authentication token:

```
vfxt.py --cloud-type aws --on-instance
```

Remote Console

If using a remote system to run vfxt.py, you must pass user credentials for an authorized user to the cloud system where you are creating vFXT clusters. This user must be associated with the cluster manager role or policy, as described in Create a User with the Cluster Manager Role, above.

Use the key pair to authenticate as the user:

```
vfxt.py --cloud-type aws --access-key access_key --secret-key secret_key
```

If you have configured the AWS command-line utility (awscli) with your credentials, you can use the `--from-environment` option to import them into vfxt.py:

```
vfxt.py --cloud-type aws --from-environment
```

AWS Environment Options

The region and subnet arguments provide enough information to query the VPC configuration. For cross-availability-zone (XAZ) configurations, you can provide multiple subnets - each in a different zone - separated by spaces.

```
vfxt.py --cloud-type aws
        --access-key access_key
        --secret-key secret_key
        --region AWS_region
        --subnet subnet subnet2 subnet3
```

Configuring AWS GovCloud

The AWS GovCloud region has custom settings beyond the GovCloud specific region. The Amazon resource name (ARN) has a specific value for GovCloud resources (`aws-us-gov`), and the GovCloud IAM service has a custom endpoint name (`iam.us-gov.amazonaws.com`).

Use the option `--govcloud` to set both of these automatically.

```
vfxt.py --cloud-type aws
        <authentication options>
        --region gov_cloud_region
        --subnet subnet subnet2 subnet3
        --govcloud
```

If you want to set them individually, use the following commands instead of the `--govcloud` option.

- To set the GovCloud resource name: `--arn aws-us-gov`
- To set the GovCloud endpoint name: `--iam-host iam.us-gov.amazonaws.com`

Extra AWS Configuration Options

Refer to Additional AWS-Specific Options or `vfxt.py --help` for details. Extra configuration options for AWS cloud environments include:

- Security groups
- Placement group
- Tagging
- Disable encryption
- Use ephemeral disks

AWS Cluster Settings

This table shows example values that can be used when creating a vFXT cluster on Amazon Web Services. Please work with your Systems Engineer (SE) or Avere Global Support (AGS) to determine the best options.

	vfxt.py script option	**Default value**	**Other value options**
VM instance type	`--instance-type`	`r3.2xlarge`	`r3.8xlarge`
SSD type	`--data-disk-type`	`gp2`	`io1`
Node cache size	`--node-cache-size`	`1000`	Up to `8000`
Number of nodes	`--nodes`	`3`	Up to `20`
Bucket creation	`--no-corefiler`	Omit this option	`--no-corefiler`

AWS Tip - Using Additional Cloud Core Filers

When you create a vFXT cluster with a cloud core filer, Avere software automatically creates an IAM policy that allows the cluster nodes to access the bucket for the core filer. (This is true regardless of whether you created a new bucket or if you specified an existing bucket to use as the core filer at creation time.)

The policy created allows access *only* to the specific storage bucket specified when creating the cluster. If you later want to add a different cloud core filer later, you must edit the IAM policy to let the nodes access the new bucket.

To allow access to a different bucket, you will change the `resource` statements in the IAM policy so that they allow access to any bucket, instead of restricting the cluster nodes to accessing one specific bucket.

(Optionally, you can add or substitute the new bucket name instead of opening access to all connected cloud buckets, but allowing access to all buckets is easier and more flexible. Refer to AWS documentation to learn how to add a second bucket resource.)

Updating the Cluster IAM Policy

Follow these steps to revise the cluster policy to allow access to a different storage bucket.

1. Open the AWS IAM console.

2. Identify the role created for the cluster. The role name has the form `policy_avere_cluster_role_##########_cluster_name`

3. Select the role for your cluster, then click the **Edit Policy** link at the bottom of the page.

4. Find the resource lines that specify the original cloud core filer. They will have the form

   ```
    "Resource": [
                   "arn:aws:s3:::bucket_name"
   ```

   There usually are two resource statements, one like the one above, and one that includes `/*` after the bucket name.

5. Replace the bucket names in these statements with a wildcard character to allow access to all buckets. Specifically:
   - Change `arn:aws:s3:::bucket_name` to `arn:aws:s3:::*`
   - Change `arn:aws:s3:::bucket_name/*` to `arn:aws:s3:::*/*`

6. Click **Apply Policy** when done.

Quick Reference - Using vfxt.py with Google Cloud Platform

This section gives an overview of how to configure a vfxt.py installation to be able to create Avere clusters in a Google Cloud Platform environment.

Note: Read the vFXT Installation Guide for Google Cloud Platform for complete information about the configuration required to create vFXT clusters in the Amazon cloud computing environment.

Configuring the GCE environment to allow vfxt.py access includes the following steps:

- Check your project's resource quotas and request an increase if needed. (Note that quota increase requests might take a day or more to be fulfilled.)
- Create a Google Compute instance
- Send project information to Avere Systems to gain access to the vFXT image
- Configure a Service Account
    - If using a remote console to access the GCE system, export the default service account credential as a JSON key file for authenticating the remote console to the GCE instance.

      Optionally, you can create and use a new service account instead of using the default account. The steps are the same - export the credential as a JSON key file.

    - If using a GCE cloud instance to run vfxt.py on the GCE system, you will obtain an on-instance authentication token instead of using a JSON key. This token will use the service account credentials the instance was created with (either the default service account or a separate account that you created for creating the cloud instance).

- Install the Google API Python client library to allow Python to interact with the GCE API.

Install the Google API Library

On the system where you will run vfxt.py, install the Google API Python Client Library.

```
sudo pip install --upgrade google-api-python-client
```

GCE Authentication Options

When issuing a vfxt.py command on a Google Cloud system, you must include the appropriate parameters to authenticate the system running vfxt.py to your Google Cloud Project.

Some steps are necessary ahead of time. If running vfxt.py from a remote console, you must download a key file associated with a service account. If running fvxt.py from a cloud console, you must make sure the instance used has full API privileges.

Cloud Instance

If using a GCE instance to run vfxt.py, create an instance that has full access to all cloud APIs. This enables API-based read and write access for compute and storage resources.

To authorize the instance, vfxt.py can query the instance metadata to obtain an authentication token. Pass the `--on-instance` value to indicate that vfxt.py should obtain a token locally.

```
vfxt.py --cloud-type gce --on-instance
```

Note that Google Cloud Shell cannot be used to run vfxt.py, since it runs in a temporary container and does not have the required IP connectivity to the cloud project.

Remote Console

If using a remote system to run vfxt.py, you must obtain and supply a JSON key file from the GCE service account. Follow the key management instructions in the Google Cloud Service Accounts documentation to learn how to download the JSON key file from the service account.

Use the `--key-file` option to provide the file to authenticate the remote system.

```
vfxt.py --cloud-type gce --key-file key_file.json
```

Note: Older Google service accounts provided .p12 PKCS#12 files instead of JSON key files. If your service account uses .p12, also use the `--client-email` option to specify the service account email associated with the key.

If you have configured the gcloud utility with your credentials, you can use the `--from-environment` option to import them into vfxt.py.

> **Note:** You must issue the command `gcloud auth application-default login` to authorize the local application to use your user credentials for API access before invoking vfxt.py.

```
vfxt.py --cloud-type gce --from-environment
```

Note: Use gcloud auth application-default login to configure credentials before invoking vfxt.py.

GCE Environment Options

For each vfxt.py command, you must pass environment options that describe your cloud project network configuration.

The `network` and `zone` options are required for all actions. These values describe where in your cloud space the vFXT clusters are (or will be created). For cross-zone configurations, you can provide multiple zones in a space-separated list.

Optionally, use the --subnetwork option to specify a subnet.

```
vfxt.py --cloud-type gce
        --key-file key_file.json
        --network network
        --zone zone zone2 zone3
        --subnetwork subnetwork
```

Additional GCE Configuration Options

Refer to the list of commands in GCE Options or `vfxt.py --help` for details about other Google-specific parameters.

Extra GCE-specific options include:

- Instance tagging
- Instance metadata
- Local SSD disks

GCE Cluster Settings

This table shows example values that can be used when creating a vFXT cluster on Google Cloud. Please work with your Systems Engineer (SE) or Avere Global Support (AGS) to determine the best options.

	vfxt.py script option	**Default value**	**Other value options**
VM instance type	`--instance-type`	`n1-highmem-8`	`n1-highmem-32`
SSD type	`--data-disk-type`	`pd-ssd` (persistent)	`local-ssd`
Node cache size	`--node-cache-size`	`1000` (for persistent disks) `750` (for local disks, in multiples of 375)	persistent disks: `4000` (min `250`, max `8000`; low numbers are intended for test configurations) local disks: `1500` (max `3000`)
Number of nodes	`--nodes`	`3`	Up to `20`
Bucket creation	`--no-corefiler`	Omit this option	`--no-corefiler`

Troubleshooting

If you have trouble using the vfxt.py script, check the following settings and messages before contacting Avere Global Services for support.

Common Issues

Common errors include the following situations:

- Insufficient cloud quotas
- Faulty permissions/role configuration
- Misconfigured network settings that do not allow access to cloud API endpoints

Debugging Help

The `--debug` option can be added to any vfxt.py command. It gives verbose feedback that can help you to identify useful status messages or detect problems in backend interaction that come about during the script's execution.

To simplify output collection, you can use the tee command to direct the standard error and output streams to a log file:

```
vfxt.py <script arguments> 2>&1 | tee logfile.txt
```

Save the debugging output from any unresolved situations to send to Avere Global Services.

Contacting Avere Global Services

Support is available from Avere Global Services.

When contacting Avere Global Services, please include the output of the script invocations that are failing.

You can reach Avere Global Services in several ways:

Telephone

US and Canada toll free: 1-888-88-AVERE (888-882-8373), press 2 for Support

United Kingdom: +44 20 3695 1638, press 2 for Support

US or International: +1-412-894-2570, press 2 for Support

Web

http://www.averesystems.com/support

Email

Email Avere Global Services at support@averesystems.com